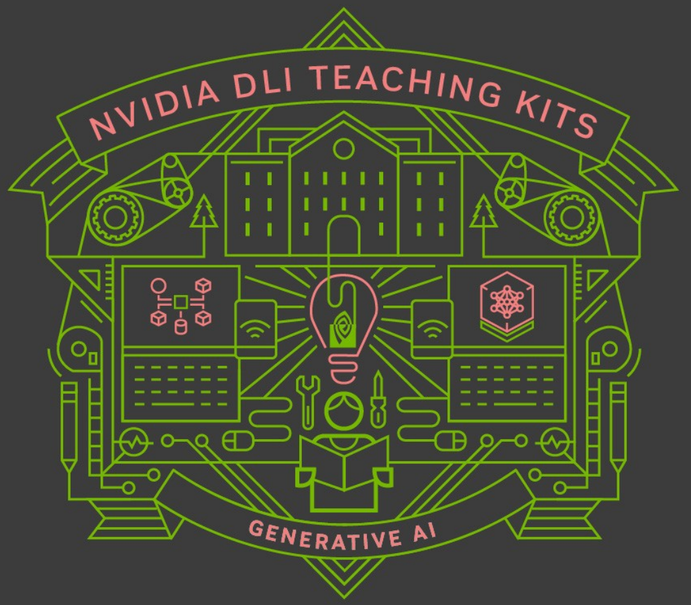
Lecture 6.1- Image Generation and GANs

Generative AI Teaching Kit



The NVIDIA Deep Learning Institute Generative AI Teaching Kit is licensed by NVIDIA and Dartmouth College under the Creative Commons Attribution-NonCommercial 4.0 International License.

This lecture

- Image Generation in the context of Deep Learning and AI
- Generating Images
- The Convolution and Transpose Convolution
- Generative Adversarial Networks (GANs)
- Pitfalls and Improvements with traditional GANs, DCGANs
- Applications of GANs, StyleGAN, CycleGAN, Conditional GANs

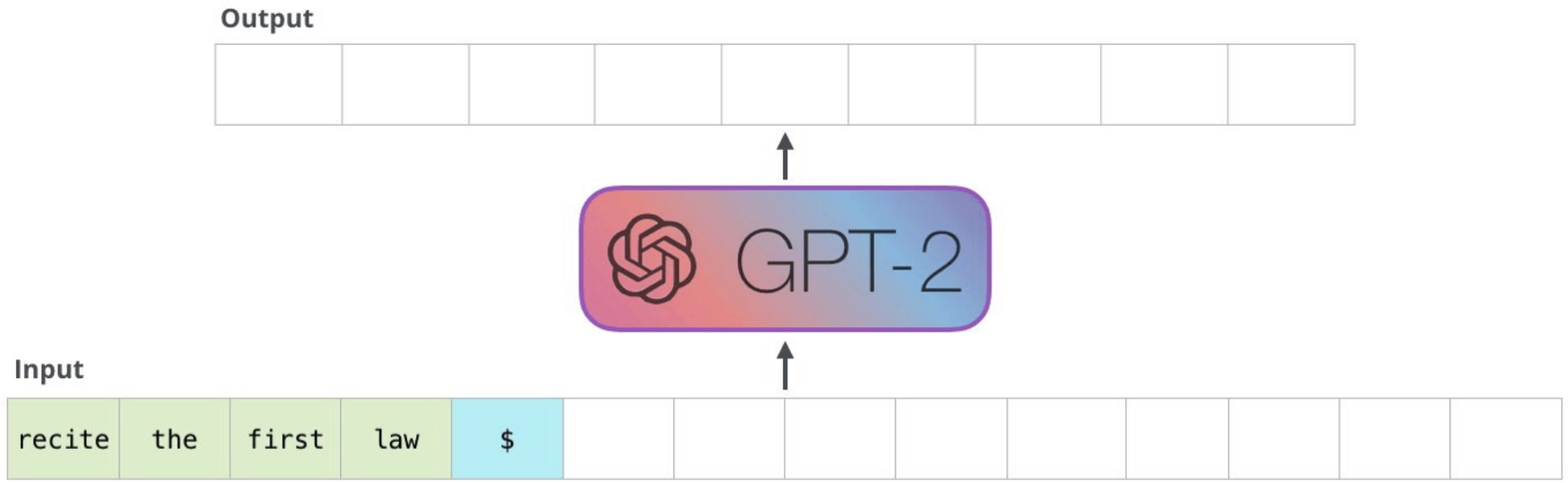
DARTMOUTH ENGINEERING | NVIDIA

Images & GenAI

Turning our attention to a different medium

Learning to speak, next to see?

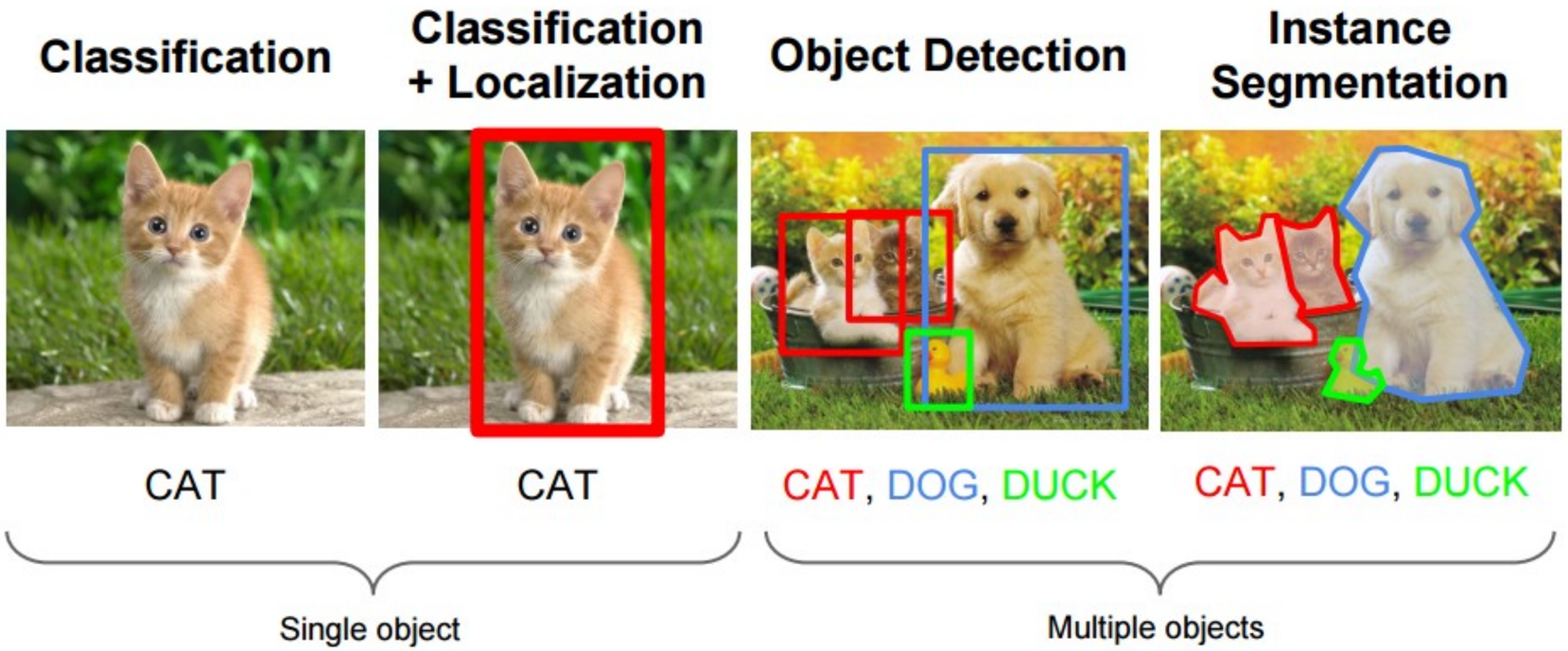
So far in this course we have focused on only text generation with LLMs as the mode of GenAI applications…



But! What about other areas of deep learning beyond Natural Language Processing?

Computer Vision

Prior to the LLM craze, Computer Vision dominated the deep learning world



Whereas Language Models attempt to model and understand human language, vision models focus on interpreting and classifying images (and videos)

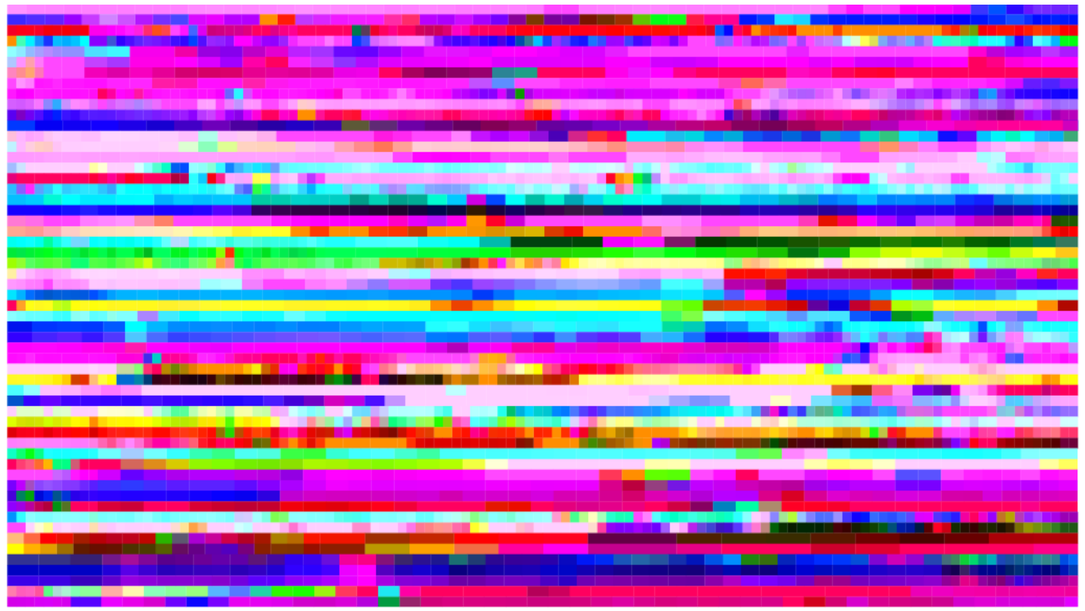
However, until recently, **generating** images was no simple task. Instead the research focused on image comprehension.

A difficult vision

✓ Language Generation

☐ Image Generation?...what is so hard about image generation?

Language has a finite and discrete vocabulary that language models can select from to generate the next token



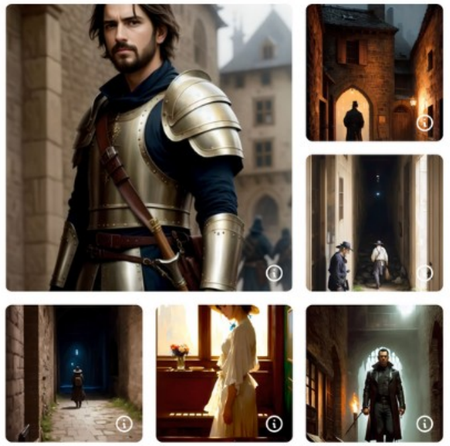
Images have a continuous multidimensional space from which we can sample, ie an infinite space to search.

DARTMOUTH ENGINEERING | NVIDIA

Why generate images in the first place?

GENERAL

Use cases

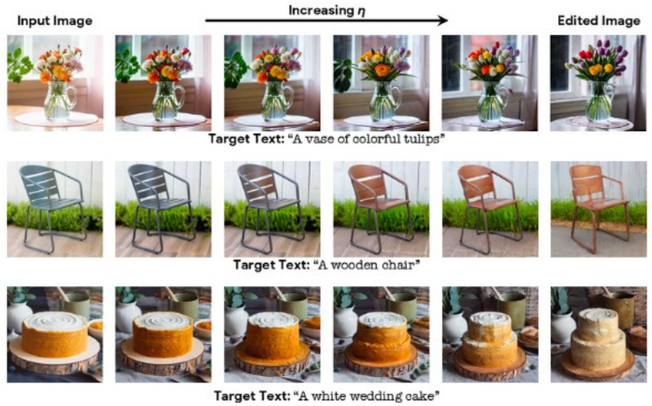


VFX

Character creation

eCommerce

Product images



Creative

Image editing

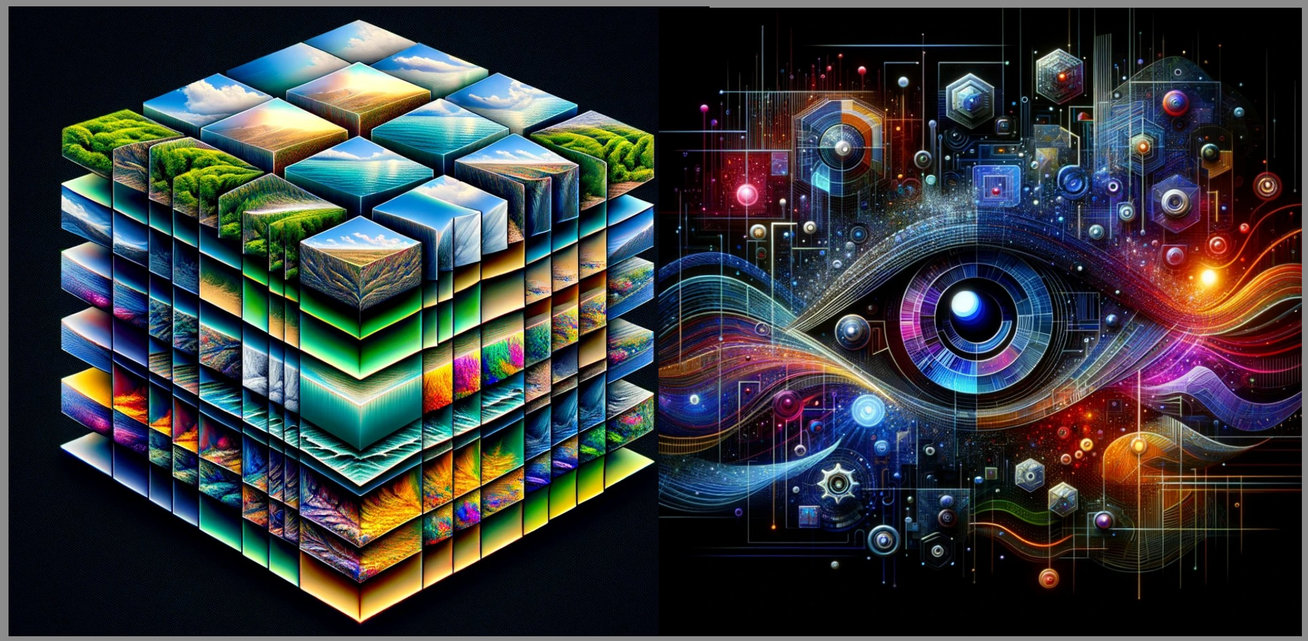
voltaML

We've still barely scratched the surface of what image generators can do

DARTMOUTH ENGINEERING | NVIDIA

Convolution Layers

How computers see and how they can draw

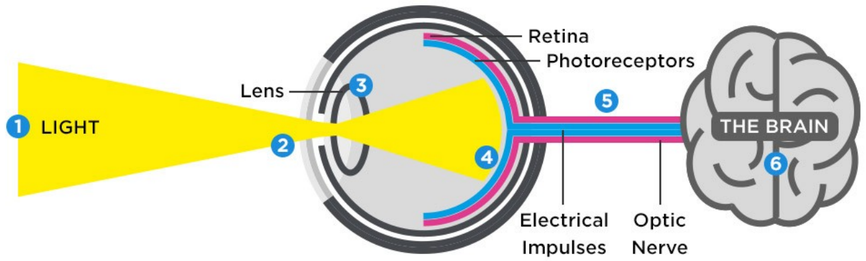


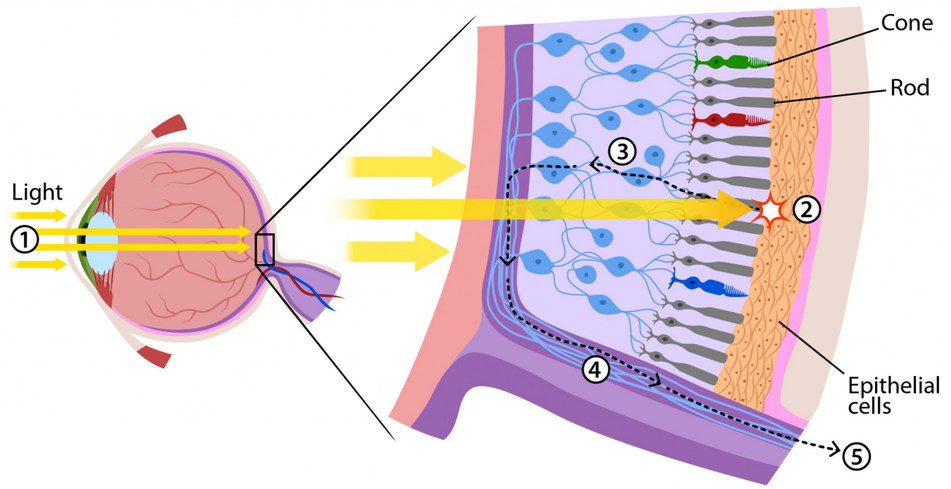
DARTMOUTH ENGINEERING | NVIDIA

How does sight work?

Rods: Since they are more sensitive to light, rods are responsible for vision at low light levels (scotopic vision). They do not mediate color vision, and have a low spatial acuity.

Cones: Cones are active at higher light levels (photopic vision), are capable of color vision, and are responsible for high spatial acuity.

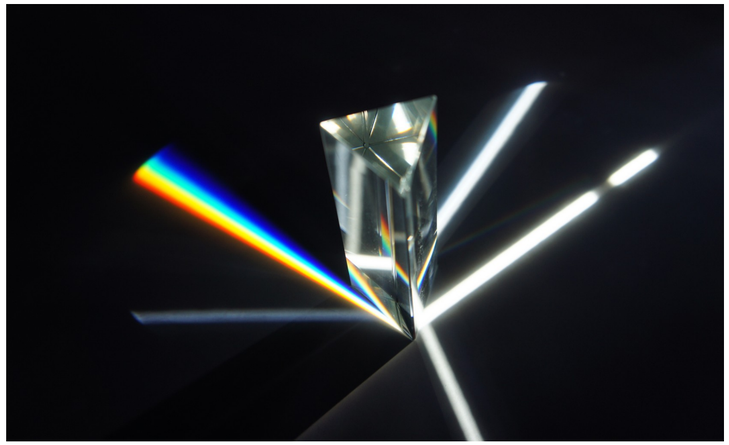


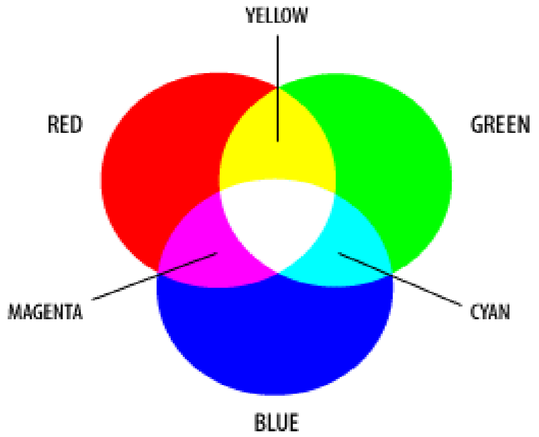


What is a digital image?

Newton’s prism experiment showed that white light is comprised of all colors.

We can represent any visible color as a combination of red, green, and blue.



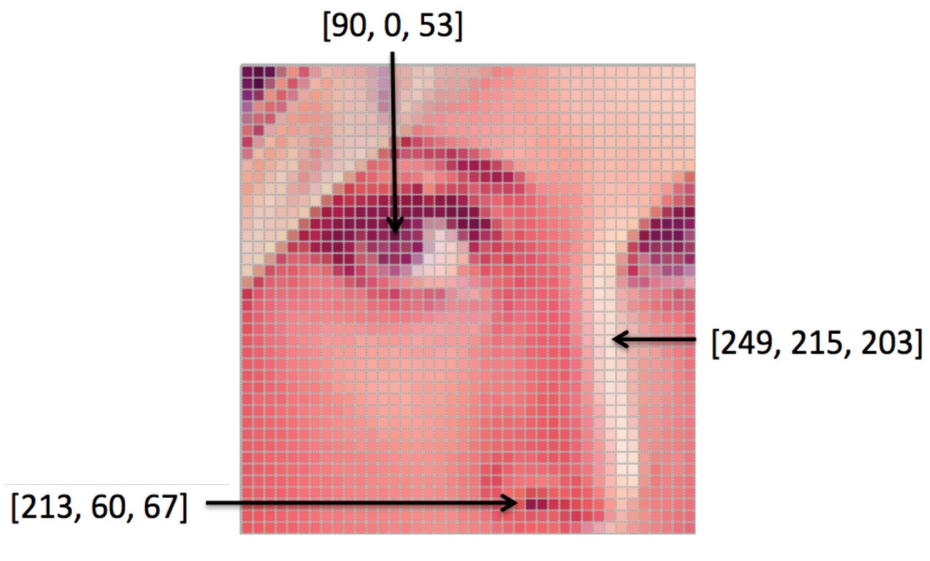


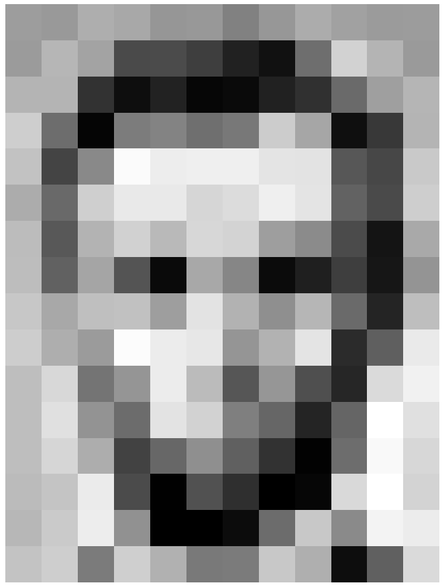
RGB Values for Common Colors

R	G	B	
255	0	0	
0	69	100	
255	255	0	
0	255	0	
0	0	255	
153	51	255	

Using numbers to see

Using the RGB ‘color space’ we can store images as a matrix.





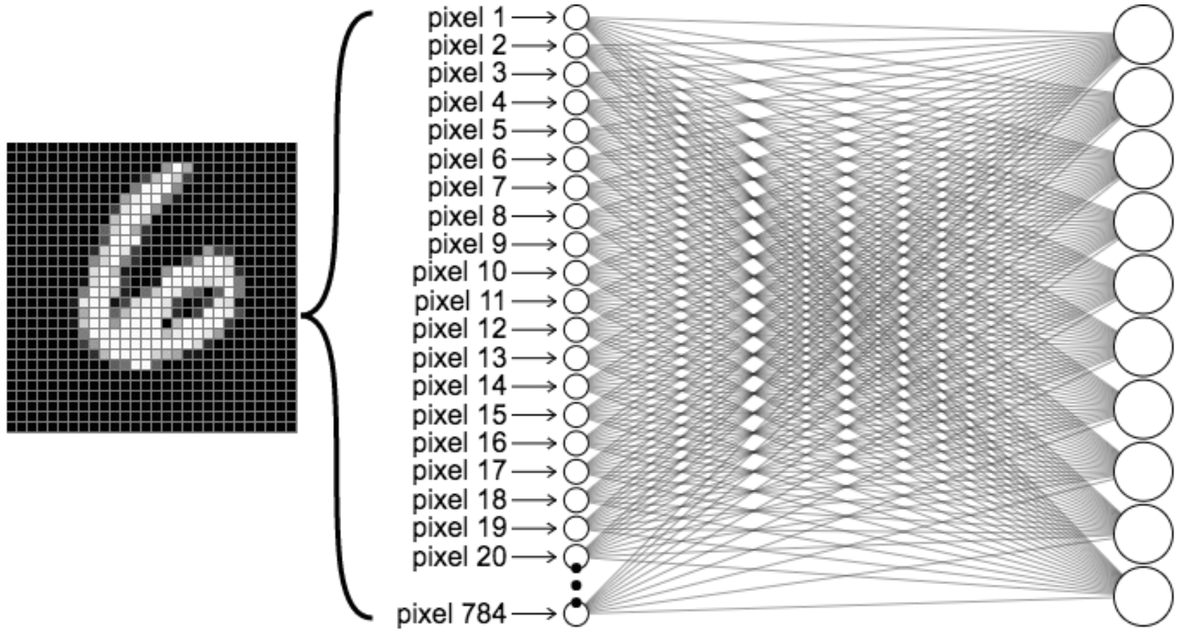
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Images but not images

MNIST

The MNIST database (**M**odified **N**ational **I**nstitute of **S**tandards and **T**echnology database) is a large database of handwritten digits that is commonly used for training various image processing systems since 1994.



Prior to the convolution layer, images like MNIST were flattened to a list of pixel values.

This method did not work for the ImageNet competition…. why?

Spatial Information

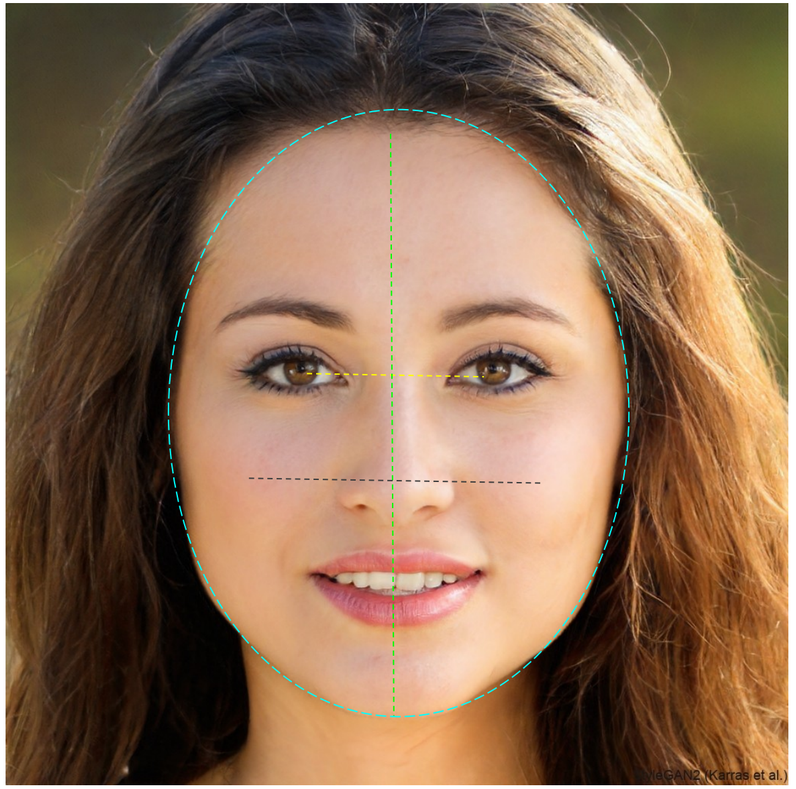
Simple neural networks see inputs independently.

Images, however, have a lot of local spatial dependency/local connectivity.

Think about facial recognition:

- Distance between the eyes
- Distance from the forehead to the chin
- Distance between the nose and mouth
- Depth of the eye sockets
- Shape of the cheekbones
- Contour of the lips, ears, and chin

We need something that can process ***spatial information***

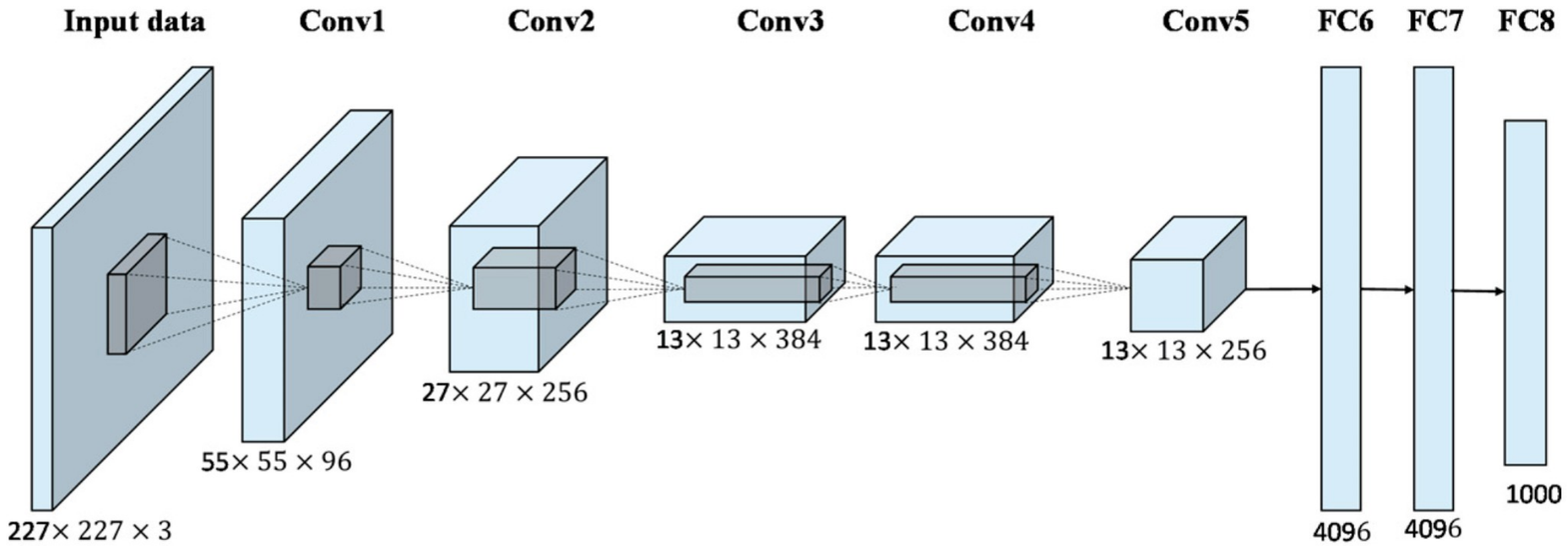


https://thispersondoesnotexist.com/

The CNN revolution - AlexNet

The neural network AlexNet won the 2012 ImageNet competition

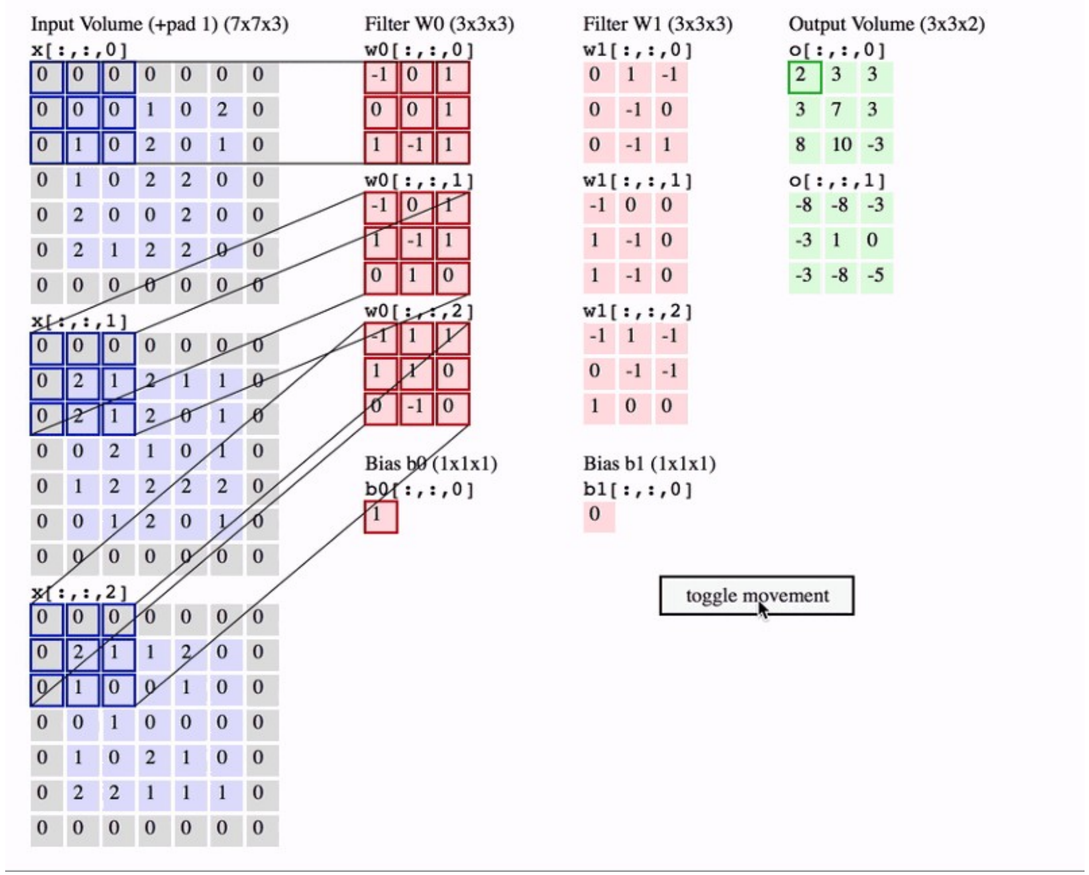
This was the first mainstream utilization of a Convolutional Neural Network and would shape the next decade of AI research and development.



AlexNet won with a top-5 error rate of 15.3%, compared to the second-place top-5 error rate of 26.2%.

Secret to success: The Convolution Operation

The *convolution* operation is the key to computer vision, and allows for image generation too



A convolution layer uses a set of **kernels** to perform **convolutions** over the image.

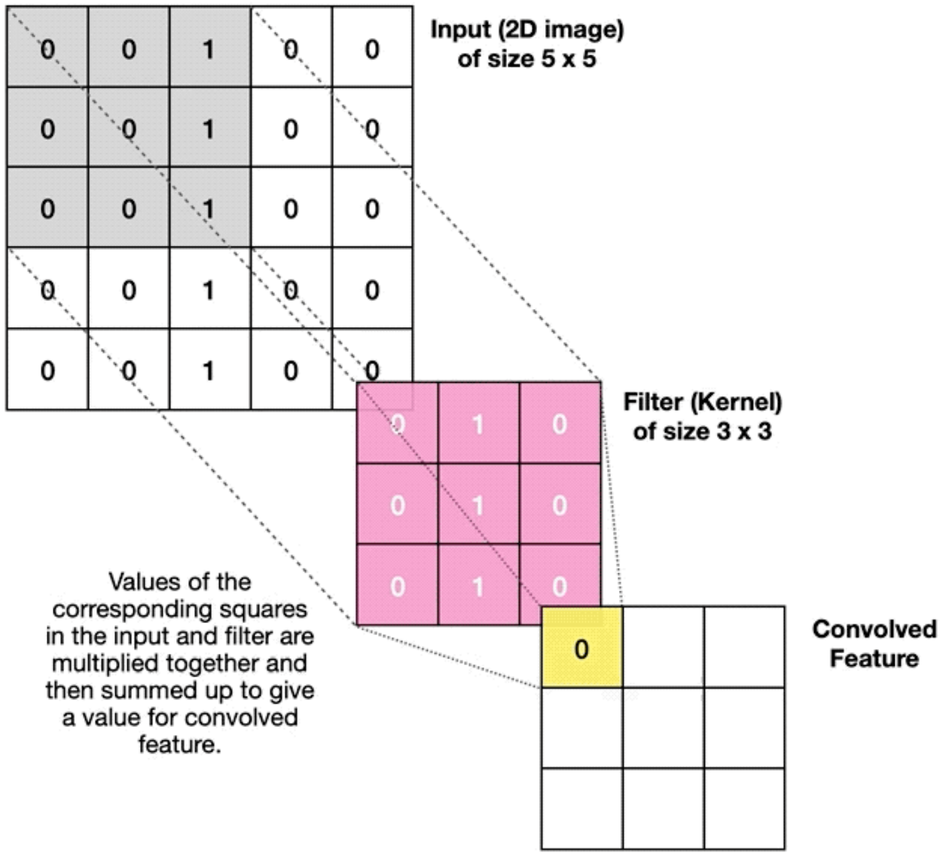
Typical sizes are 3x3 and 5x5. The kernels are **learned** in training.

Being able to span vertical and horizontal directions to gather information, these kernels can interpret spatial information very effectively and efficiently.

DARTMOUTH ENGINEERING | NVIDIA

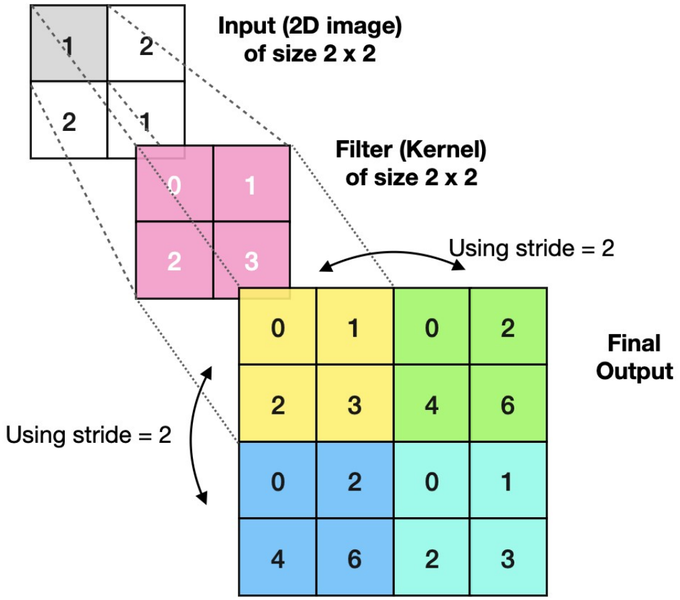
Going the other direction?

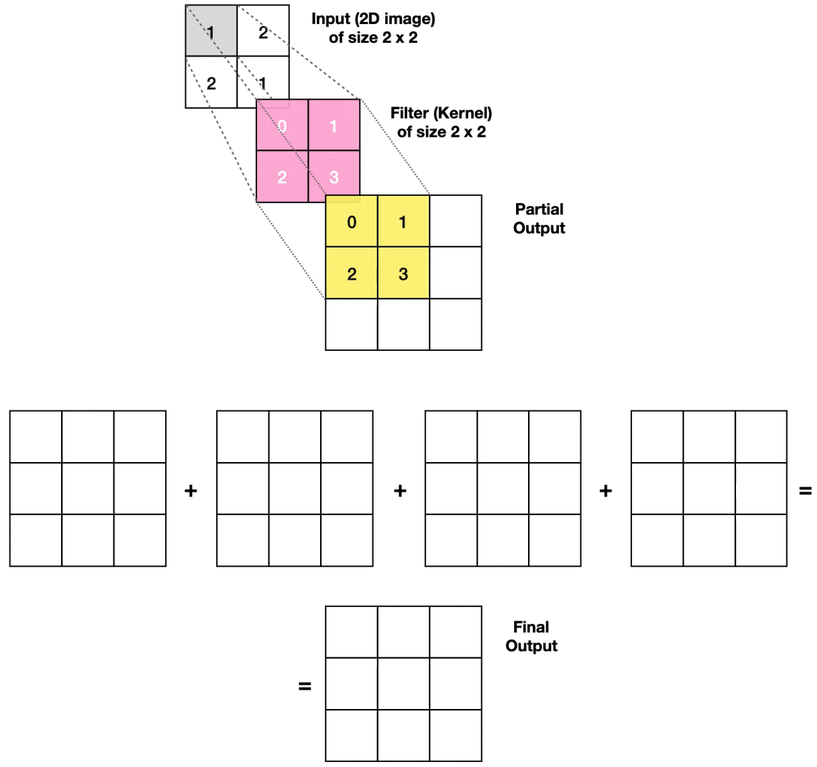
- Convolution Layers are used to distill parts of the image into a lower dimensional representation of the image
- Information is lost at each state of a convolution, so a direct reversal would not generate the same image
- But if we allow for some “fuzziness” in the generated image, we can flip the flow of information to expand the dimension of the data



The Transpose Convolution

- The process involves using a kernel and a stride to slide over the input tensor.
- Transpose convolutions do not combine multiple input values into a single output (as in standard convolution).
- Instead, it spreads a single input value across multiple output positions, effectively increasing the spatial dimensions.





Generative Adversarial Networks

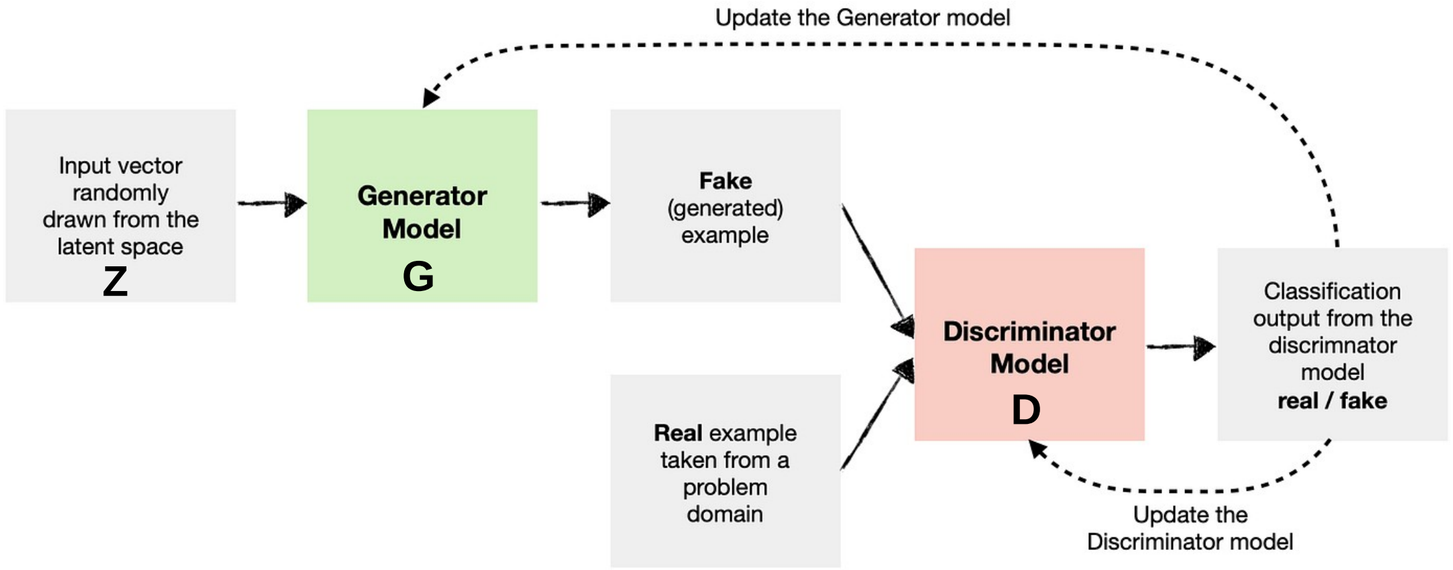
Generative AI before “GenAI”

Generative Adversarial Networks - What is a GAN?

The GAN was originally proposed by Ian Goodfellow in 2014 in which two neural networks contend against each other, hence “adversarial”

The Generator model, **G**, takes as input a random vector, **Z**, and is trained to generate some data.

The Discriminator model, **D**, is a classifying whose sole role is to determine if a sample came from the training dataset, or is a fake generated by **G**.



When trained, we can smoothly vary the input vector **Z** to generate new samples

Generative Adversarial Networks – Training a GAN

Training GANs couples the Generator and Discriminator losses

The generator generates synthetic data from random noise, $z \sim p_z(z)$
Where $p_z(z)$ is a prior distribution (e.g., uniform or normal distribution).

The discriminator outputs a probability $D(x)$ indicating whether input, x , is real or fake.

$D(x)$ is trained to maximize the probability of correctly classifying real data $x \sim p_{data}(x)$, and synthetic data, $G(z)$

Generator Loss

$$\mathcal{L}_G = -\mathbb{E}_{z \sim p_z(z)}[\log D(G(z))]$$

The Generator tries to maximize the log-likelihood of synthetic data being classified as real.

Discriminator Loss

$$\mathcal{L}_D = -\mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] - \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

The Discriminator tries to maximize the log-likelihood of real data being classified as real and synthetic data being classified as fake.

From the original GAN paper

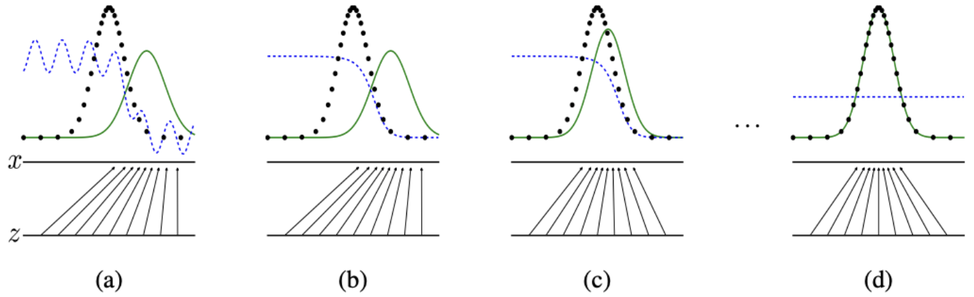


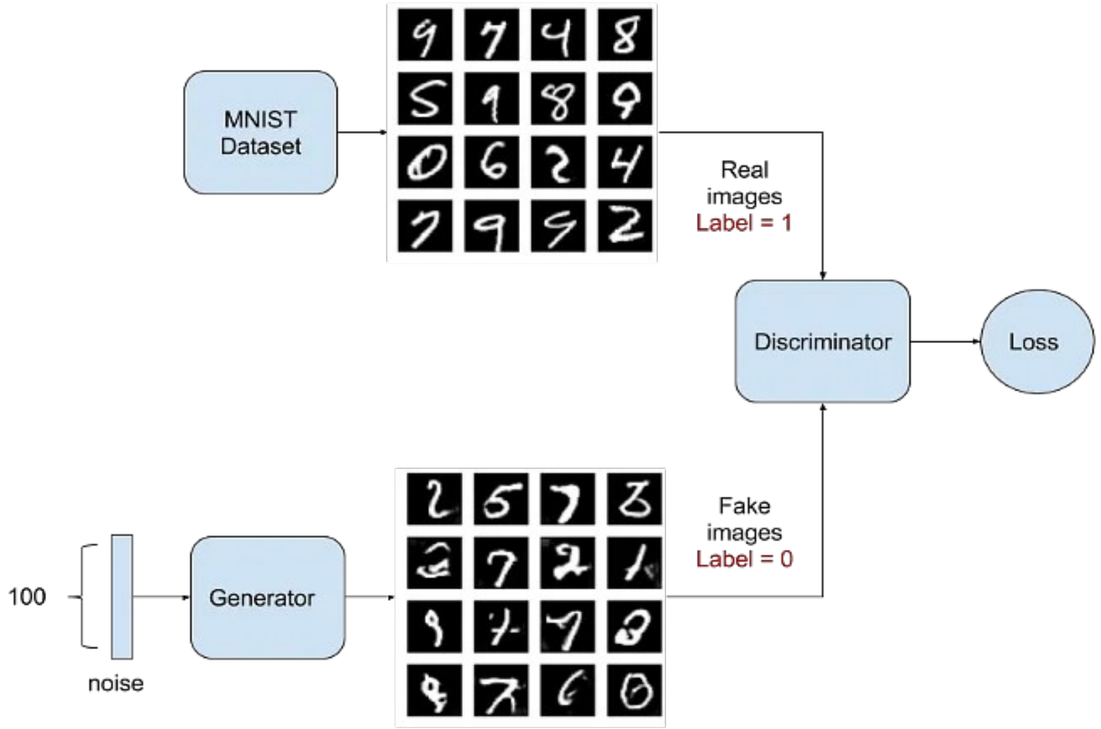
Figure 1: Generative adversarial nets are trained by simultaneously updating the **d**iscriminative distribution (D, blue, dashed line) so that it discriminates between samples from the data generating distribution (black, dotted line) $p_{\boldsymbol{x}}$ from those of the **g**enerative distribution p_g (G) (green, solid line). The lower horizontal line is the domain from which $\boldsymbol{z}$ is sampled, in this case uniformly. The horizontal line above is part of the domain of $\boldsymbol{x}$. The upward arrows show how the mapping $\boldsymbol{x} = G(\boldsymbol{z})$ imposes the non-uniform distribution p_g on transformed samples. G contracts in regions of high density and expands in regions of low density of p_g. (a) Consider an adversarial pair near convergence: p_g is similar to p_{data} and D is a partially accurate classifier. (b) In the inner loop of the algorithm D is trained to discriminate samples from data, converging to $D^*(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$. (c) After an update to G, gradient of D has guided $G(\boldsymbol{z})$ to flow to regions that are more likely to be classified as data. (d) After several steps of training, if G and D have enough capacity, they will reach a point at which both cannot improve because $p_g = p_{\text{data}}$. The discriminator is unable to differentiate between the two distributions, i.e. $D(\boldsymbol{x}) = \frac{1}{2}$.

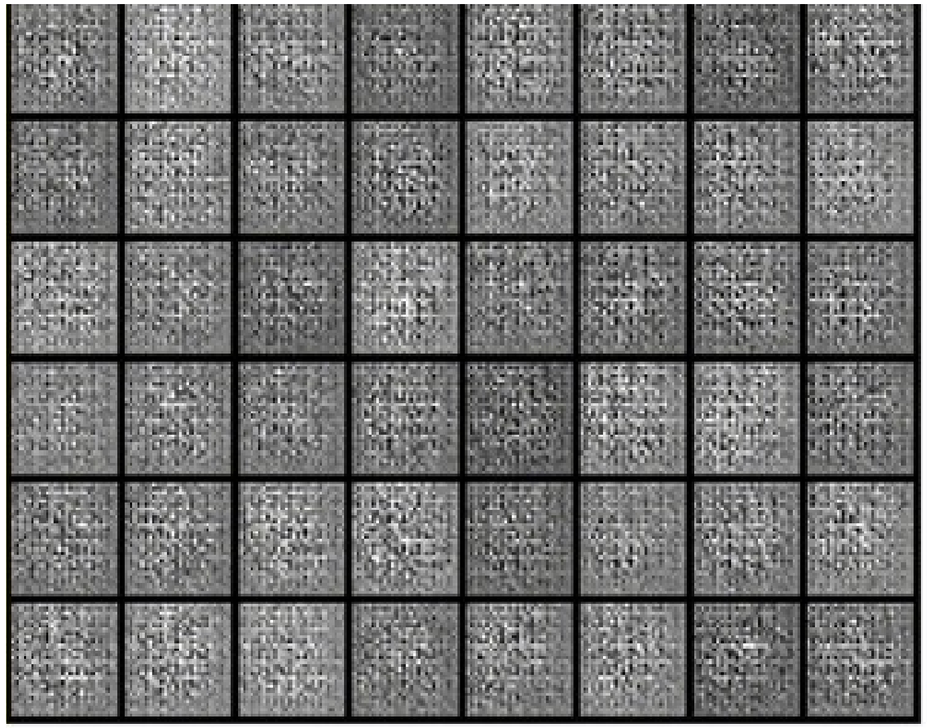
DARTMOUTH ENGINEERING | NVIDIA

Generative Adversarial Networks – Creating Images

GANs can generate essentially any form of data.

For the MNIST problem, GANs can produce a flat array of 0s and 1s.





Progression of generating MNIST examples as a GAN is trained

Generative Adversarial Networks – Deep Convolution GANs

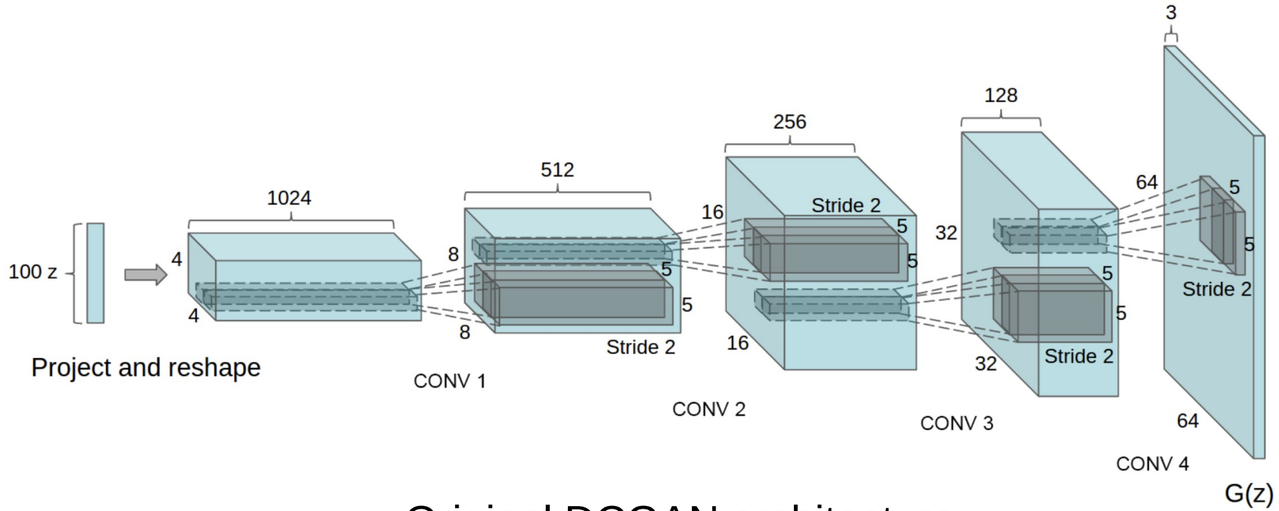
Recall that we have seen the transpose convolution. We can create a new type of generator, one that includes these transpose convolution.

DCGAN (**D**eep **C**onvolution **GAN**)

Input a random column vector

Passed through several transpose convolutions

Final result is a generated image!

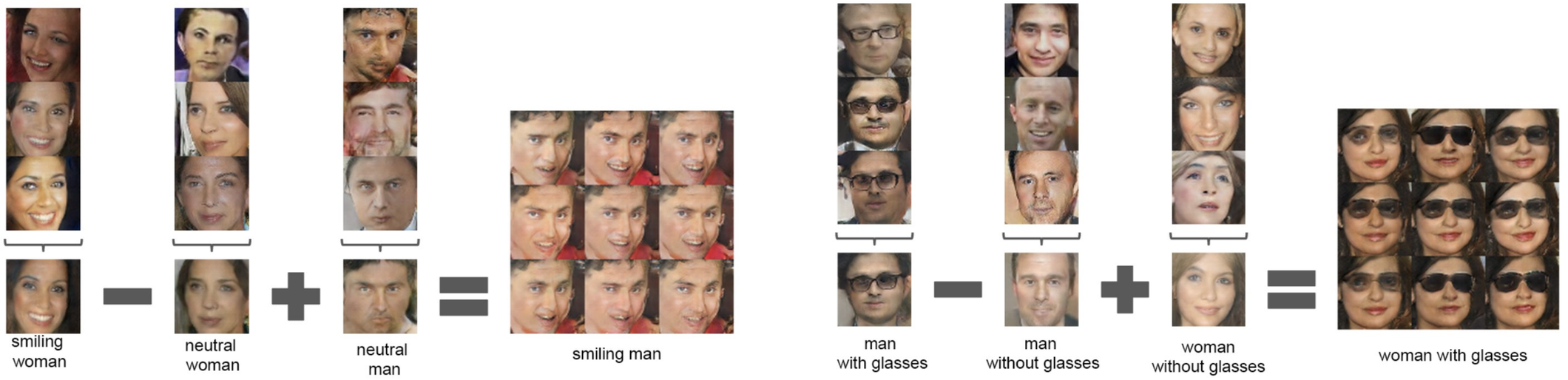


Original DCGAN architecture

DCGAN – Exploring the latent space of images

With a trained DCGAN, we can manipulate the latent vector, z, to explore the space of images we can generate.

Latent Vector Space (adding the z vectors)



Pixel Space (adding the pixel values)

DCGAN – Exploring the latent space of images

By carefully controlling how we change the latent vector, we can even reliably control the image changes.

Here the authors discovered how to induce a turn in the image subject by finding a “turn vector to add to the latent vector

DARTMOUTH ENGINEERING | NVIDIA

Improvements and Challenges of GANs

Training GANs can be especially adversarial…

Generative Adversarial Networks – Issues and Challenges

Training Instability:

GANs are notoriously difficult to train, often suffering from issues like mode collapse and vanishing gradients.

Hyperparameter Sensitivity:

The performance of GANs is highly sensitive to the choice of hyperparameters, requiring extensive tuning.

Lack of Interpretability:

GANs are considered black-box models, making it hard to understand the internal workings and learned features.

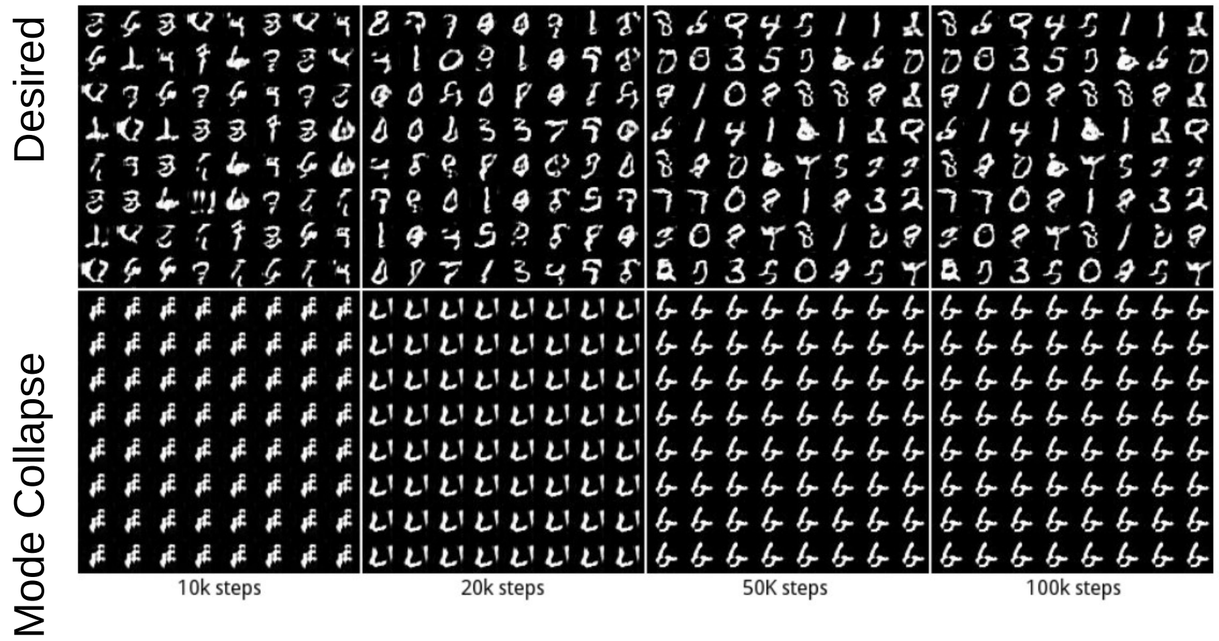
DARTMOUTH ENGINEERING | NVIDIA

Generative Adversarial Networks – Mode Collapse

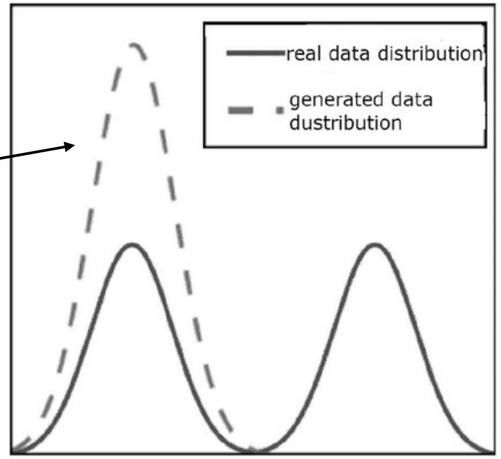
During training a Generator can find a local minima which produces strong feedback from the discriminator.

After this point, there is little or no impetus for the Generator to try new examples.

This collapse into a single output is known as **Mode Collapse**



Generator gets stuck in a subset of the distribution that passes the discriminator

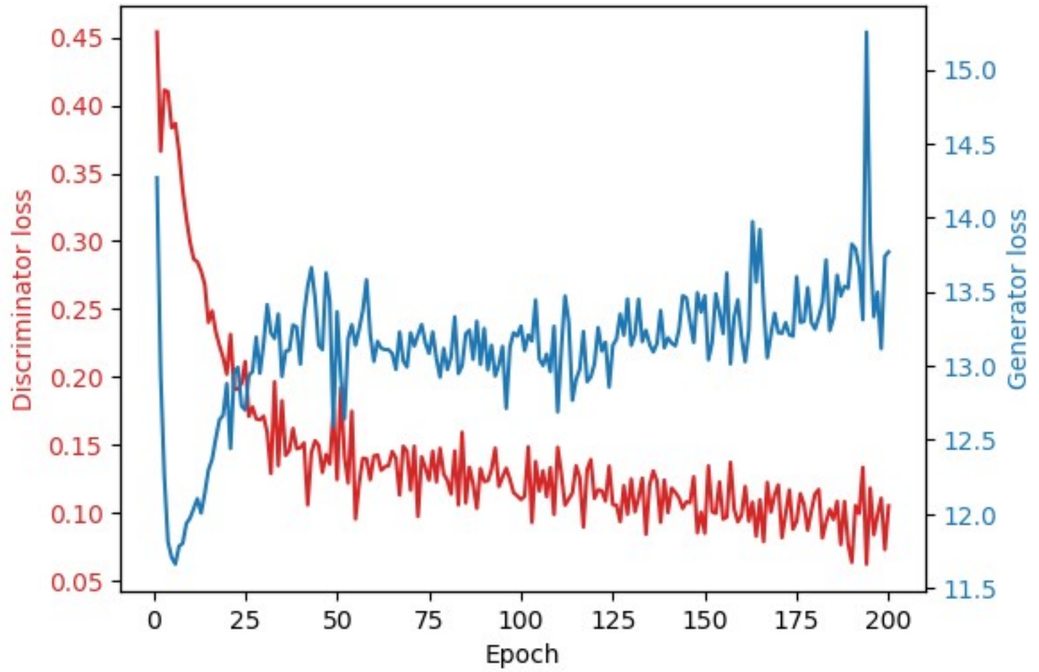


DARTMOUTH ENGINEERING | NVIDIA

Generative Adversarial Networks – Instability

Another challenge of GAN training is **instability**, consider the following:

- Generator improvement leads to discriminator decline.
- Perfect generator results in 50% discriminator accuracy (coin flip).
- Convergence issue: discriminator feedback loses value over time.
- Continued training on random feedback degrades generator quality.



If the Discriminator is too strong, then the Generator struggles to train.

Conversely, if the Discriminator is too weak, not enough information for updates is sent to the Generator.

In both cases, the Generator will be suboptimal and the training unstable.

DARTMOUTH ENGINEERING | NVIDIA

Modern development of GANs

Latest state-of-the-art approaches

DARTMOUTH ENGINEERING | NVIDIA

Modern GAN Approaches

Modern improvements to GANs have focused on:

- Improving training stability.
- Enhancing output quality.
- Addressing mode collapse issues.

Innovations such as **StyleGAN** introduced finer control over generated images.

Conditional GANs expanded the scope of GAN applications by incorporating conditional data.

CycleGANs enabled unpaired image-to-image translation, broadening the usability of GANs in diverse domains.

Let’s take a look at these new approaches.

DARTMOUTH ENGINEERING | NVIDIA

Modern GAN Approaches - StyleGAN

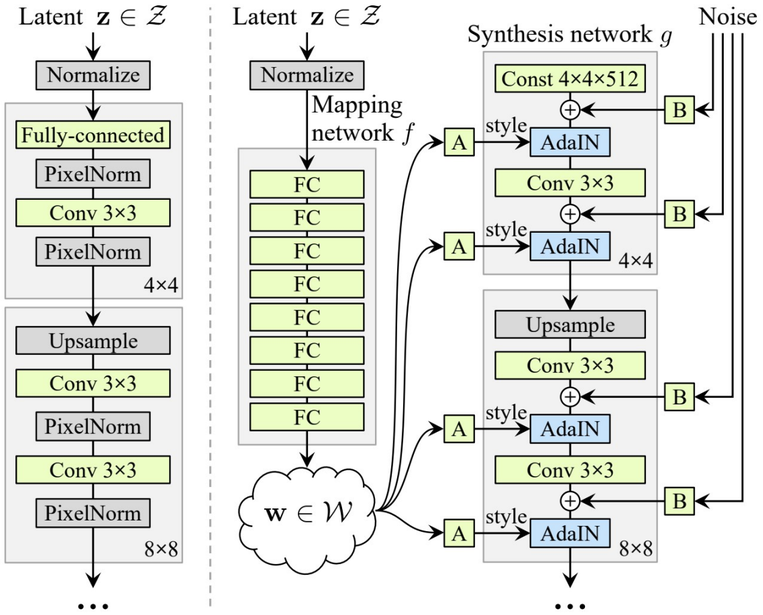
StyleGAN, developed by NVIDIA, this is a pioneering GAN architecture for generating high-quality, photorealistic images.

Unlike traditional GANs, StyleGAN introduces a style-based generator architecture that provides fine-grained control over various aspects of the generated image, from overall structure to detailed textures.

Style Control: Uses a mapping network to transform the input noise vector into an intermediate latent space, allowing manipulation of image styles at different levels.

Progressive Growing: Gradually increases the resolution of the generated images during training, enhancing stability and quality.

High-Quality Outputs: Produces highly realistic images, making it ideal for applications in art, fashion, and virtual reality.



(a) Traditional (b) Style-based generator

StyleGAN3 Mixing of styles

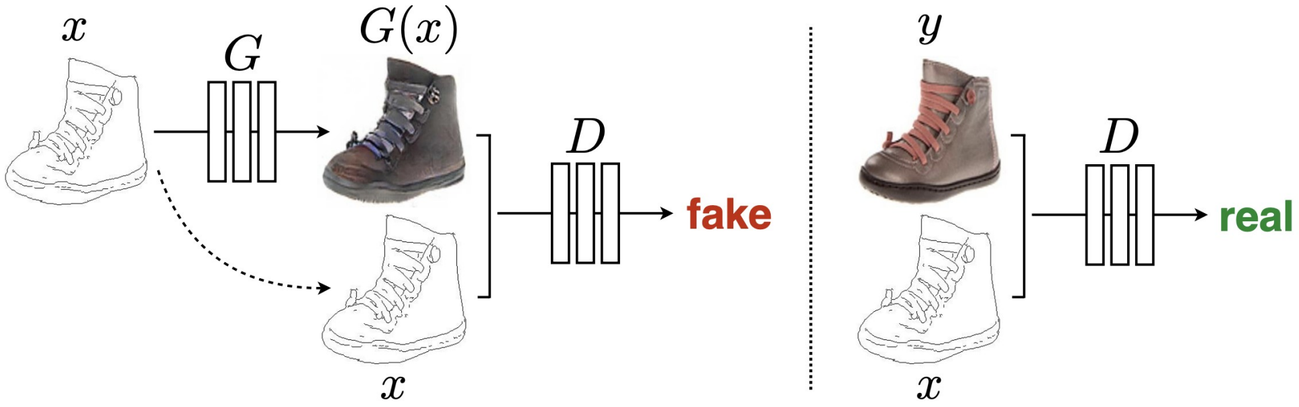
DARTMOUTH ENGINEERING | NVIDIA

Modern GAN Approaches – Conditional GANs

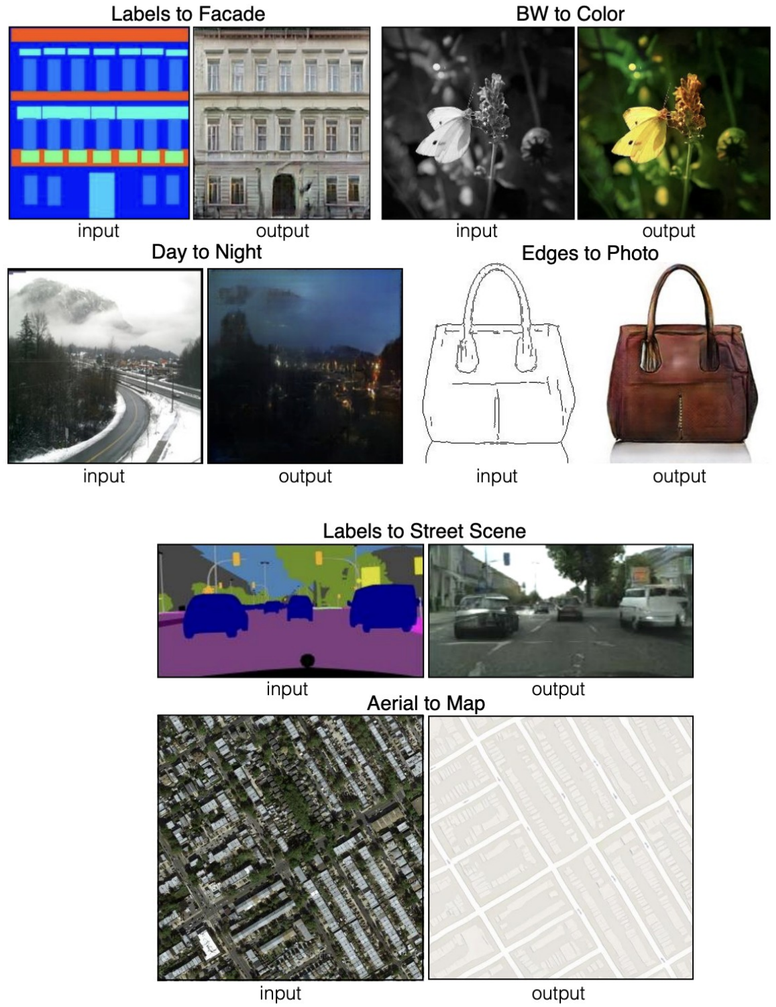
Conditional GAN (cGAN): Targeted Image Generation

cGANs extend the standard GAN framework by conditioning the generation process on additional information, such as class labels or specific attributes.

This approach allows for more controlled and targeted image generation, enabling the creation of specific outputs based on the given conditions.



Conditioning Information: Both the Generator and Discriminator receive additional input data (e.g., class labels), guiding the generation process.



Modern GAN Approaches - CycleGANs

CycleGAN: Unpaired Image-to-Image Translation

CycleGAN is designed for unpaired image-to-image translation, allowing the transformation of images from one domain to another without needing paired training examples.

Cycle Consistency Loss: Ensures that an image translated from one domain to another and back remains unchanged, promoting stable and accurate translations.

Dual GANs: Uses two GANs to learn the mapping between the two domains, ensuring consistency and quality in both directions.

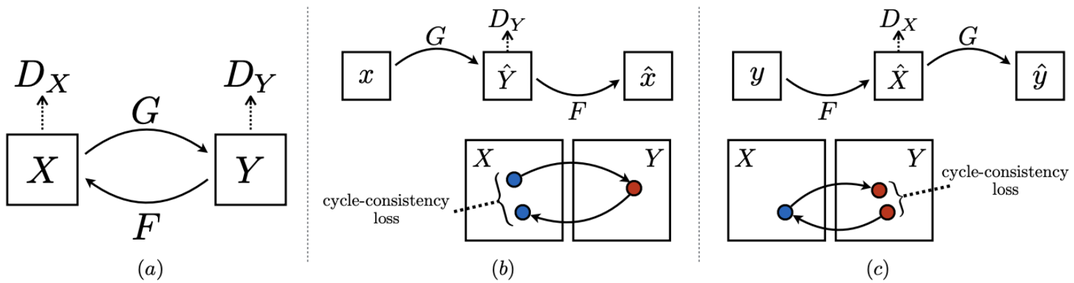
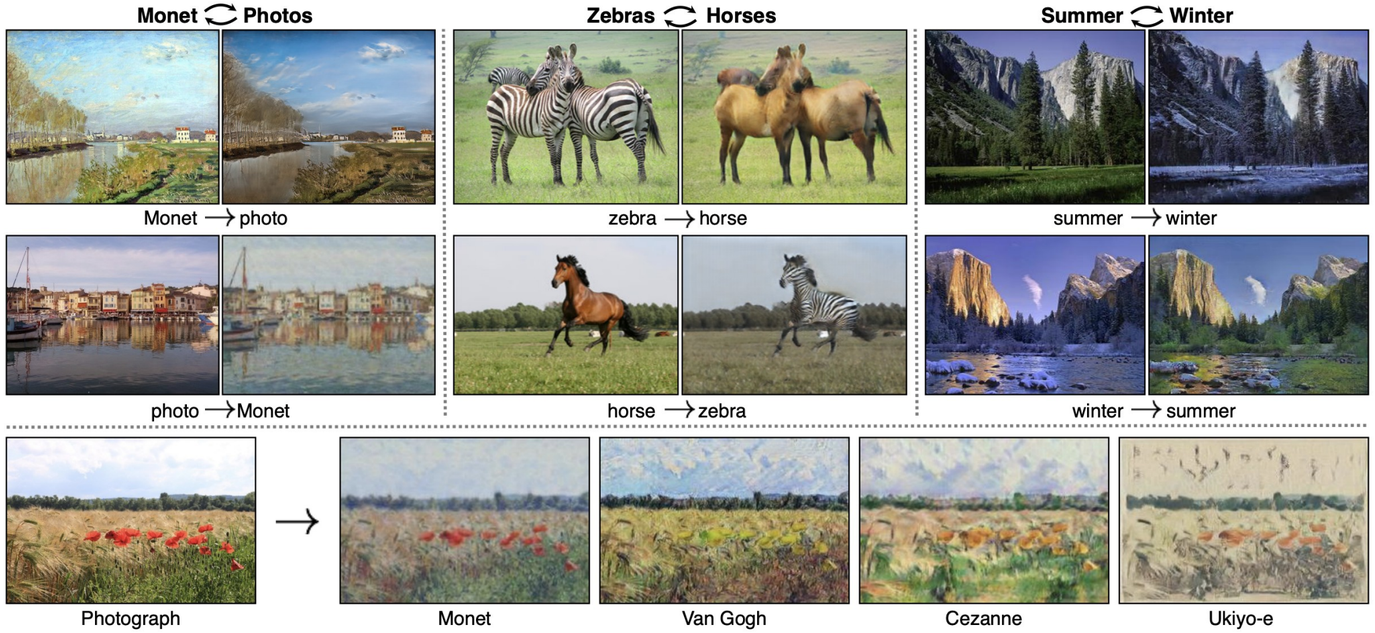


Figure 3: (a) Our model contains two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and associated adversarial discriminators D_Y and D_X. D_Y encourages G to translate X into outputs indistinguishable from domain Y, and vice versa for D_X and F. To further regularize the mappings, we introduce two *cycle consistency losses* that capture the intuition that if we translate from one domain to the other and back again we should arrive at where we started: (b) forward cycle-consistency loss: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$, and (c) backward cycle-consistency loss: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$



Wrap Up

Computer Vision and Generative AI

- Today we reviewed some important computer vision topics
- Introduced the transpose convolution and how it can be used to generate images
- Explored Generative Adversarial Networks (GANs)
- Discussed the issues with training GANs including mode collapse and instability
- Saw some of the latest developments in GANs: StyleGAN, cGAN, CycleGAN

In the next lesson we will start our discussion on diffusion models and how they can overcome some of the GAN limitations.

DARTMOUTH ENGINEERING | NVIDIA

Thank you!